

# On the Development & Performance of a First Order Stokes Finite Element Ice Sheet Dycore Built Using *Trilinos* Software Components

**I. Tezaur\***, A. Salinger, M. Perego, R. Tuminaro

Sandia National Laboratories  
Livermore, CA and Albuquerque, NM

*With contributions from: I. Demeshko (SNL),  
S. Price (LANL) and M. Hoffman (LANL)*

Saturday, March 14, 2015

SIAM Conference on Computational Science  
& Engineering (CS&E) 2015  
Salt Lake City, UT

\*Formerly I. Kalashnikova.

SAND2015-1599 C

# Outline

- The First Order Stokes model for ice sheets and the *Albany/FELIX* finite element solver.
- Verification and mesh convergence.
- Effect of partitioning and vertical refinement.
- Nonlinear solver robustness.
- Linear solver scalability.
- Performance-portability.
- Summary and ongoing work.



# Outline

- The First Order Stokes model for ice sheets and the *Albany/FELIX* finite element solver.
- Verification and mesh convergence.
- Effect of partitioning and vertical refinement.
- Nonlinear solver robustness.
- Linear solver scalability.
- Performance-portability.
- Summary and ongoing work.

## **For non-ice sheet modelers, this talk will show:**

- How one can rapidly develop a production-ready scalable and robust code using open-source libraries.
- Recommendations based on numerical lessons learned.
- New algorithms / numerical techniques.



# The First-Order Stokes Model for Ice Sheets & Glaciers

- Ice sheet dynamics are given by the **“First-Order” Stokes PDEs**: approximation\* to viscous incompressible **quasi-static** Stokes flow with power-law viscosity.

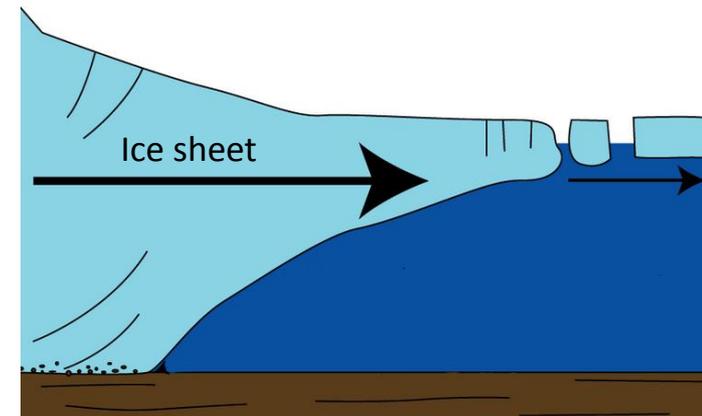
$$\begin{cases} -\nabla \cdot (2\mu \dot{\epsilon}_1) = -\rho g \frac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu \dot{\epsilon}_2) = -\rho g \frac{\partial s}{\partial y} \end{cases}, \quad \text{in } \Omega$$

$$\begin{aligned} \dot{\epsilon}_1^T &= (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13}) \\ \dot{\epsilon}_2^T &= (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23}) \\ \dot{\epsilon}_{ij} &= \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \end{aligned}$$

- Viscosity  $\mu$  is nonlinear function given by **“Glen’s law”**:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left( \frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}^2 \right)^{\left( \frac{1}{2n} - \frac{1}{2} \right)} \quad (n = 3)$$

- Relevant boundary conditions:



\***Assumption**: aspect ratio  $\delta$  is small and normals to upper/lower surfaces are almost vertical.

# The First-Order Stokes Model for Ice Sheets & Glaciers

- Ice sheet dynamics are given by the **“First-Order” Stokes PDEs**: approximation\* to viscous incompressible **quasi-static** Stokes flow with power-law viscosity.

$$\begin{cases} -\nabla \cdot (2\mu \dot{\epsilon}_1) = -\rho g \frac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu \dot{\epsilon}_2) = -\rho g \frac{\partial s}{\partial y} \end{cases}, \quad \text{in } \Omega$$

$$\begin{aligned} \dot{\epsilon}_1^T &= (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13}) \\ \dot{\epsilon}_2^T &= (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23}) \\ \dot{\epsilon}_{ij} &= \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \end{aligned}$$

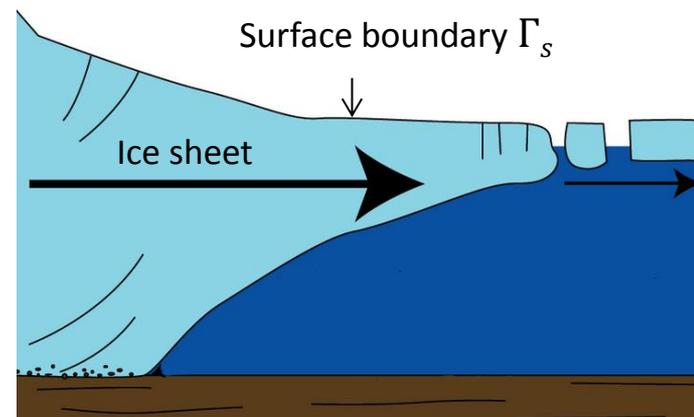
- Viscosity  $\mu$  is nonlinear function given by **“Glen’s law”**:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left( \frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}^2 \right)^{\left( \frac{1}{2n} - \frac{1}{2} \right)}$$

( $n = 3$ )

- Relevant boundary conditions:

- Stress-free BC:**  $2\mu \dot{\epsilon}_i \cdot \mathbf{n} = 0$ , on  $\Gamma_s$



\***Assumption:** aspect ratio  $\delta$  is small and normals to upper/lower surfaces are almost vertical.

# The First-Order Stokes Model for Ice Sheets & Glaciers

- Ice sheet dynamics are given by the **“First-Order” Stokes PDEs**: approximation\* to viscous incompressible **quasi-static** Stokes flow with power-law viscosity.

$$\begin{cases} -\nabla \cdot (2\mu \dot{\epsilon}_1) = -\rho g \frac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu \dot{\epsilon}_2) = -\rho g \frac{\partial s}{\partial y} \end{cases}, \quad \text{in } \Omega$$

$$\begin{aligned} \dot{\epsilon}_1^T &= (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13}) \\ \dot{\epsilon}_2^T &= (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23}) \\ \dot{\epsilon}_{ij} &= \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \end{aligned}$$

- Viscosity  $\mu$  is nonlinear function given by **“Glen’s law”**:

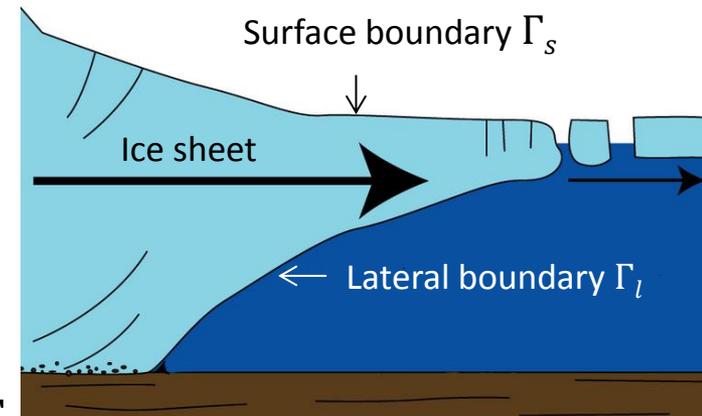
$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left( \frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}^2 \right)^{\left( \frac{1}{2n} - \frac{1}{2} \right)}$$

( $n = 3$ )

- Relevant boundary conditions:

- Stress-free BC:**  $2\mu \dot{\epsilon}_i \cdot \mathbf{n} = 0$ , on  $\Gamma_s$
- Floating ice BC:**

$$2\mu \dot{\epsilon}_i \cdot \mathbf{n} = \begin{cases} \rho g z \mathbf{n}, & \text{if } z > 0 \\ 0, & \text{if } z \leq 0 \end{cases}, \quad \text{on } \Gamma_l$$



\***Assumption:** aspect ratio  $\delta$  is small and normals to upper/lower surfaces are almost vertical.

# The First-Order Stokes Model for Ice Sheets & Glaciers

- Ice sheet dynamics are given by the **“First-Order” Stokes PDEs**: approximation\* to viscous incompressible **quasi-static** Stokes flow with power-law viscosity.

$$\begin{cases} -\nabla \cdot (2\mu \dot{\epsilon}_1) = -\rho g \frac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu \dot{\epsilon}_2) = -\rho g \frac{\partial s}{\partial y} \end{cases}, \quad \text{in } \Omega$$

$$\begin{aligned} \dot{\epsilon}_1^T &= (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13}) \\ \dot{\epsilon}_2^T &= (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23}) \\ \dot{\epsilon}_{ij} &= \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \end{aligned}$$

- Viscosity  $\mu$  is nonlinear function given by **“Glen’s law”**:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left( \frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}^2 \right)^{\left( \frac{1}{2n} - \frac{1}{2} \right)}$$

( $n = 3$ )

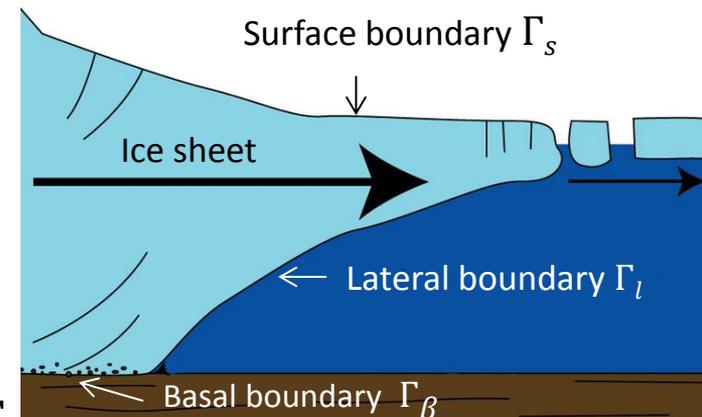
- Relevant boundary conditions:

- Stress-free BC:**  $2\mu \dot{\epsilon}_i \cdot \mathbf{n} = 0$ , on  $\Gamma_s$

- Floating ice BC:**

$$2\mu \dot{\epsilon}_i \cdot \mathbf{n} = \begin{cases} \rho g z \mathbf{n}, & \text{if } z > 0 \\ 0, & \text{if } z \leq 0 \end{cases}, \quad \text{on } \Gamma_l$$

- Basal sliding BC:**  $2\mu \dot{\epsilon}_i \cdot \mathbf{n} + \beta u_i = 0$ , on  $\Gamma_\beta$



$$\beta = \text{sliding coefficient} \geq 0$$

\*Assumption: aspect ratio  $\delta$  is small and normals to upper/lower surfaces are almost vertical.

# The PISCEES Project and the *Albany/FELIX* Solver



**“PISCEES”** = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
*5 Year Project funded by SciDAC, which began in June 2012*

**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

# The PISCEES Project and the *Albany/FELIX* Solver



“PISCEES” = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
*5 Year Project funded by SciDAC, which began in June 2012*

**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX\****.

\**FELIX*=“Finite Elements for Land Ice eXperiments”

***Albany/FELIX* Solver (steady):**  
Ice Sheet PDEs (First Order Stokes)  
(stress-velocity solve)

# The PISCEES Project and the *Albany/FELIX* Solver



**“PISCEES”** = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
*5 Year Project funded by SciDAC, which began in June 2012*

**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX\****.
- **Requirements for *Albany/FELIX*:**

\**FELIX*=“Finite Elements for Land Ice eXperiments”

***Albany/FELIX* Solver (steady):**  
Ice Sheet PDEs (First Order Stokes)  
(stress-velocity solve)

# The PISCEES Project and the *Albany/FELIX* Solver



“PISCEES” = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
*5 Year Project funded by SciDAC, which began in June 2012*

**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX\****.
- **Requirements for *Albany/FELIX*:**
  - Scalable, fast, robust.

\**FELIX*=“Finite Elements for Land Ice eXperiments”

***Albany/FELIX* Solver (steady):**  
Ice Sheet PDEs (First Order Stokes)  
(stress-velocity solve)

# The PISCEES Project and the *Albany/FELIX* Solver



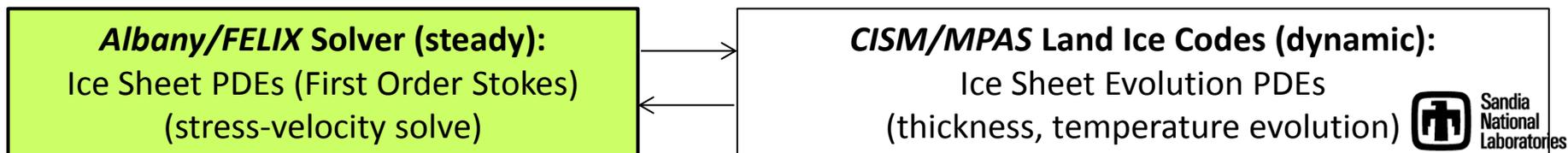
“PISCEES” = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
5 Year Project funded by SciDAC, which began in June 2012

**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX\****.
- Requirements for *Albany/FELIX*:**
  - Scalable, fast, robust.
  - Dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations (*CISM/MPAS* codes).

Dycore will provide actionable predictions of 21<sup>st</sup> century sea-level rise (including uncertainty).

\**FELIX*=“Finite Elements for Land Ice eXperiments”



# The PISCEES Project and the *Albany/FELIX* Solver



“PISCEES” = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
5 Year Project funded by SciDAC, which began in June 2012

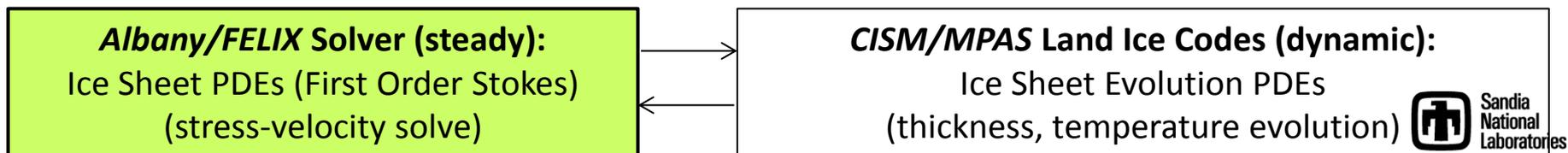
**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX***\*
- Requirements for *Albany/FELIX*:**
  - Scalable, fast, robust.
  - Dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations (*CISM/MPAS* codes).
  - Advanced analysis capabilities (deterministic inversion\*\*, Bayesian calibration, UQ, sensitivity analysis).

Dycore will provide actionable predictions of 21<sup>st</sup> century sea-level rise (including uncertainty).

\*\*See M. Perego’s talk today@ 5:25PM in MS71: “Advances on Ice-Sheet Model Initialization using the First Order Model”

\**FELIX*=“Finite Elements for Land Ice eXperiments”



# The PISCEES Project and the *Albany/FELIX* Solver



“PISCEES” = Predicting Ice Sheet Climate & Evolution at Extreme Scales  
5 Year Project funded by SciDAC, which began in June 2012

**Sandia’s Role in the PISCEES Project:** to **develop** and **support** a robust and scalable land ice solver based on the “First-Order” (FO) Stokes physics

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX***\*
- Requirements for *Albany/FELIX*:**
  - Scalable, fast, robust.
  - Dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations (*CISM/MPAS* codes).
  - Advanced analysis capabilities (deterministic inversion\*\*, Bayesian calibration, UQ, sensitivity analysis).
  - Performance-portability.

Dycore will provide actionable predictions of 21<sup>st</sup> century sea-level rise (including uncertainty).

\*\*See M. Perego’s talk today@ 5:25PM in MS71: “Advances on Ice-Sheet Model Initialization using the First Order Model”

\**FELIX*=“Finite Elements for Land Ice eXperiments”

***Albany/FELIX* Solver (steady):**  
Ice Sheet PDEs (First Order Stokes)  
(stress-velocity solve)

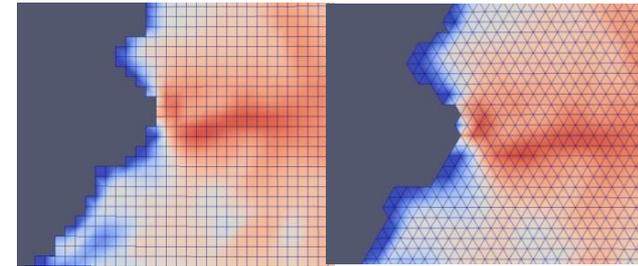
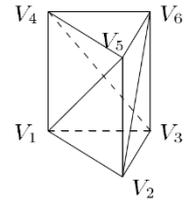
***CISM/MPAS* Land Ice Codes (dynamic):**  
Ice Sheet Evolution PDEs  
(thickness, temperature evolution)



# Algorithmic Choices for *Albany/FELIX*: Discretization & Meshes

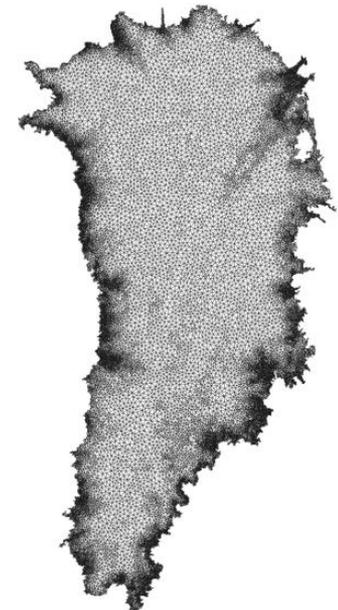
- **Discretization**: unstructured grid finite element method (FEM)

- Can handle readily complex geometries.
- Natural treatment of stress boundary conditions.
- Enables regional refinement/unstructured meshes.
- Wealth of software and algorithms.



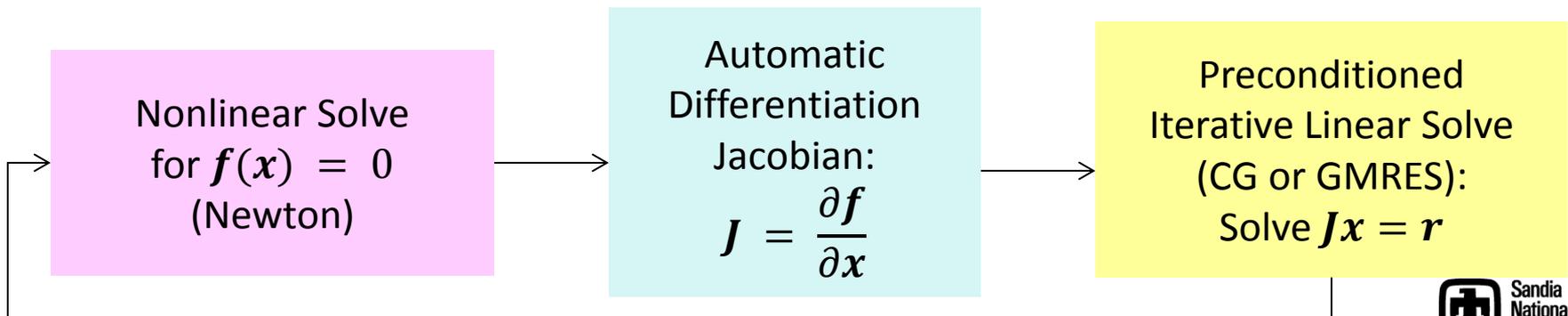
- **Meshes**: can use any mesh but interested specifically in

- ***Structured hexahedral*** meshes (compatible with *CISM*).
- ***Structured tetrahedral*** meshes (compatible with *MPAS*)
- ***Unstructured Delaunay triangle*** meshes with regional refinement based on gradient of surface velocity.
- All meshes are extruded (structured) in vertical direction as tetrahedra or hexahedra.



# Algorithmic Choices for *Albany/FELIX*: Nonlinear & Linear Solver

- **Nonlinear solver:** full Newton with analytic (automatic differentiation) derivatives
  - Most robust and efficient for steady-state solves.
  - Jacobian available for preconditioners and matrix-vector products.
  - Analytic sensitivity analysis.
  - Analytic gradients for inversion.
- **Linear solver:** preconditioned iterative method
  - **Solvers:** Conjugate Gradient (CG) or GMRES
  - **Preconditioners:** ILU or algebraic multi-grid (AMG)



# The *Albany/FELIX* Solver: Implementation in *Albany* using *Trilinos*

\*Available on github: <https://github.com/gahansen/Albany>.

The *Albany/FELIX* First Order Stokes solver is implemented in a Sandia (open-source\*) parallel C++ finite element code called...

Started  
by A.  
Salinger



Land Ice Physics Set  
(*Albany/FELIX code*)

Other Albany  
Physics Sets

## "Agile Components"

- Discretizations/meshes
- Solver libraries
- Preconditioners
- Automatic differentiation
- Many others!



- Parameter estimation
- Uncertainty quantification
- Optimization
- Bayesian inference



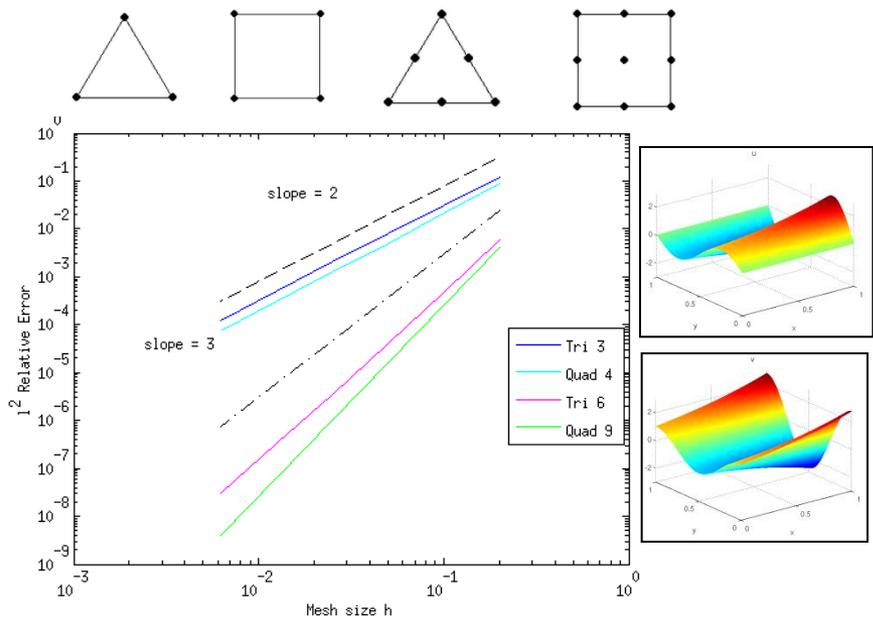
- Configure/build/test/documentation

Use of *Trilinos* components has enabled the **rapid** development of the *Albany/FELIX* First Order Stokes dycore!

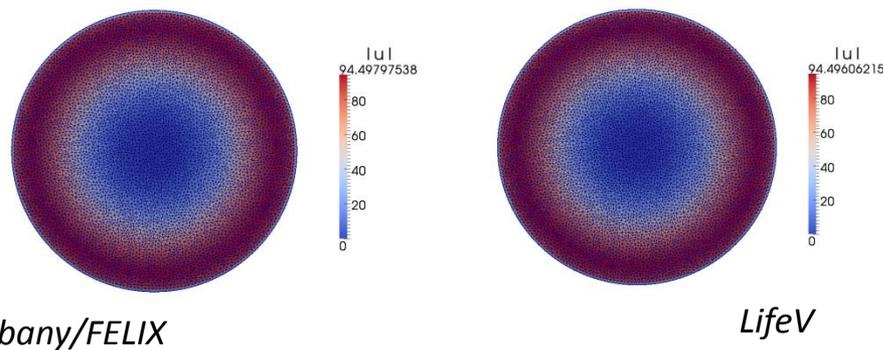
See A. Salinger's talk on Tuesday @ 2:40PM in MS225  
 "Albany: A *Trilinos*-based code for Ice Sheet Simulations and other Applications"

# Verification/Mesh Convergence Studies

**Stage 1:** solution verification on 2D MMS problems we derived.



**Stage 2:** code-to-code comparisons on canonical ice sheet problems.

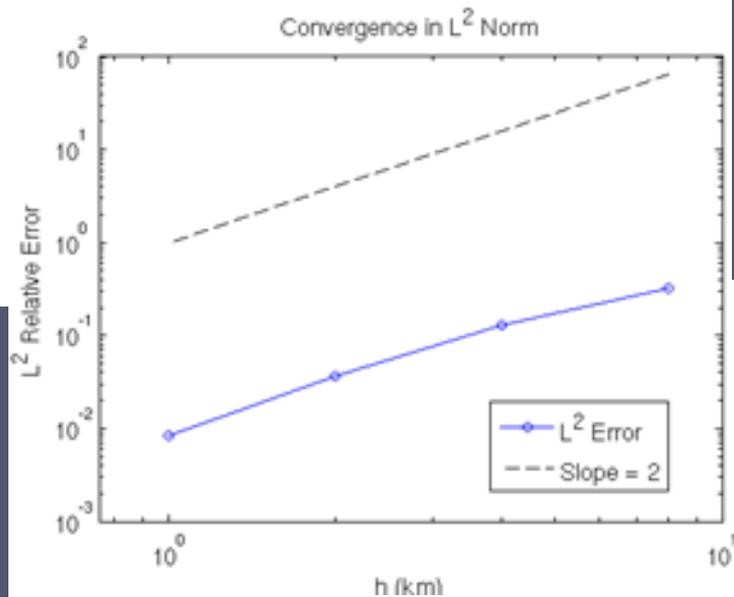


Albany/FELIX

LifeV

**Stage 3:** full 3D mesh convergence study on Greenland w.r.t. reference solution.

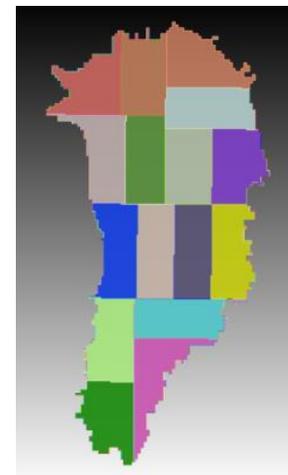
*Are the Greenland problems resolved?  
Is theoretical convergence rate achieved?*



# Mesh Partitioning & Vertical Refinement

Mesh convergence studies led to some useful practical recommendations  
(for ice sheet modelers *and* geo-scientists)!

- **Partitioning matters:** good solver performance obtained with 2D partition of mesh (all elements with same  $x, y$  coordinates on same processor - *right*).
- **Number of vertical layers matters:** more gained in refining # vertical layers than horizontal resolution (*below – relative errors for Greenland*).

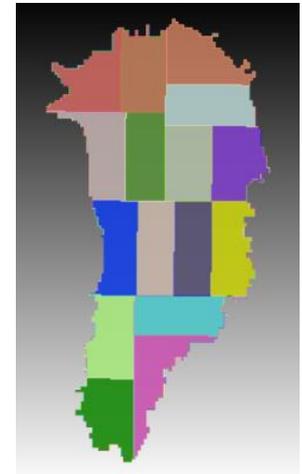


Horiz. res.\vert. layers	5	10	20	40	80
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

# Mesh Partitioning & Vertical Refinement

Mesh convergence studies led to some useful practical recommendations  
(for ice sheet modelers *and* geo-scientists)!

- **Partitioning matters:** good solver performance obtained with 2D partition of mesh (all elements with same  $x, y$  coordinates on same processor - *right*).
- **Number of vertical layers matters:** more gained in refining # vertical layers than horizontal resolution (*below – relative errors for Greenland*).

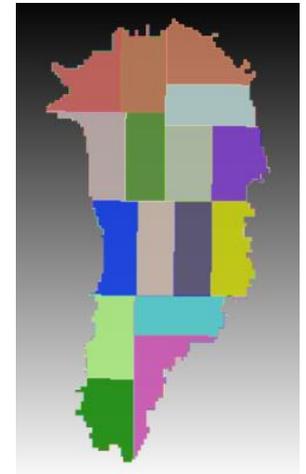


Horiz. res. \ vert. layers	5	10	20	40	80
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

# Mesh Partitioning & Vertical Refinement

Mesh convergence studies led to some useful practical recommendations  
(for ice sheet modelers *and* geo-scientists)!

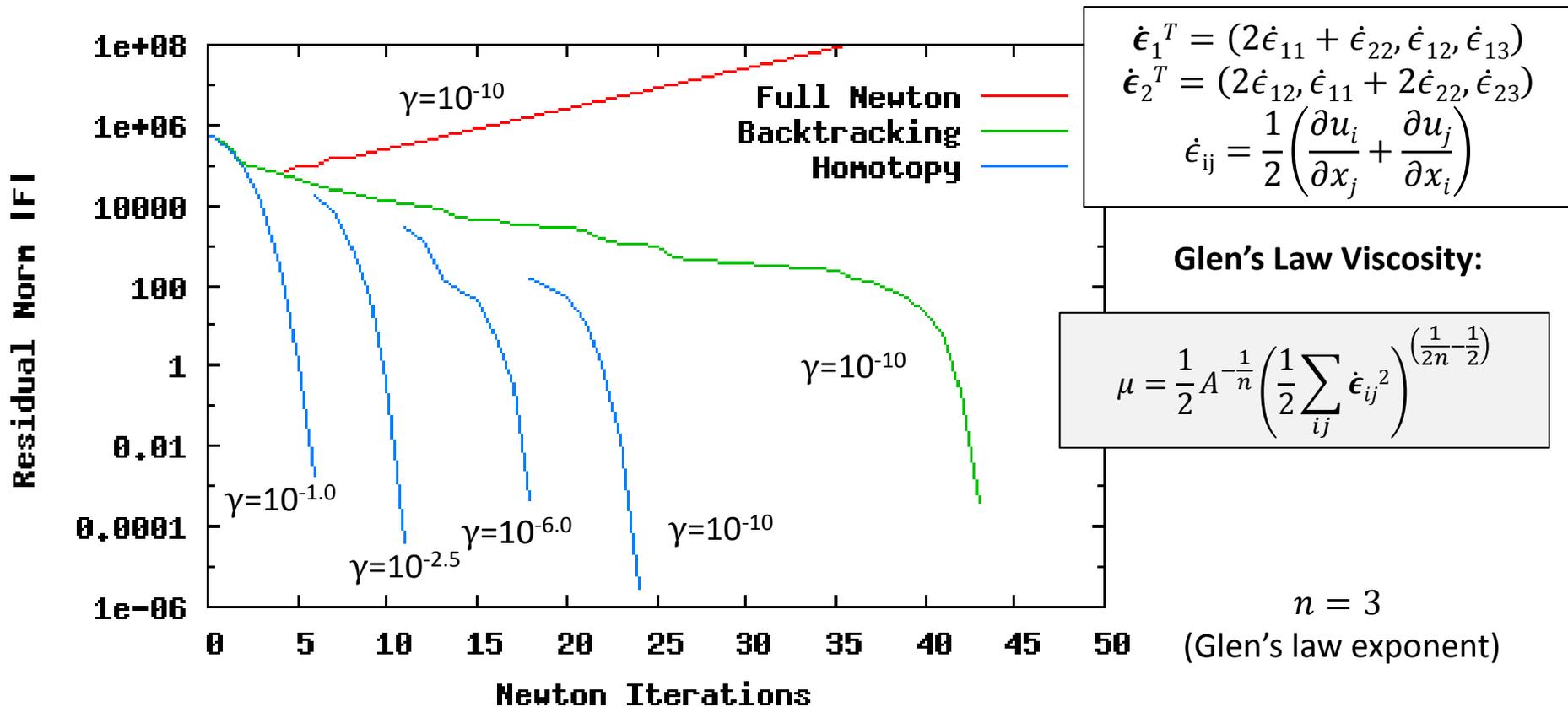
- **Partitioning matters:** good solver performance obtained with 2D partition of mesh (all elements with same  $x, y$  coordinates on same processor - *right*).
- **Number of vertical layers matters:** more gained in refining # vertical layers than horizontal resolution (*below – relative errors for Greenland*).



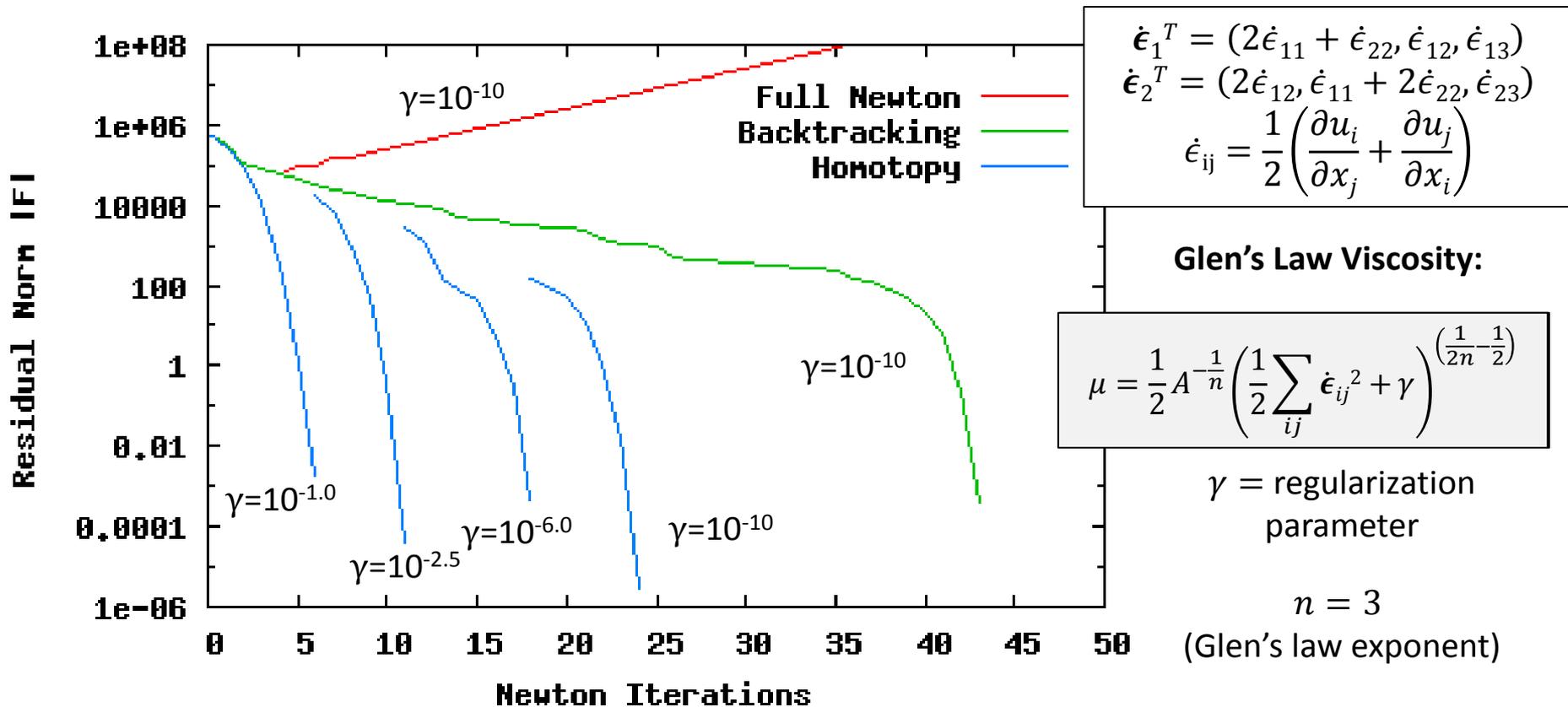
Horiz. res.\vert. layers	5	10	20	40	80
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

Vertical refinement to 20 layers recommended for 1km resolution over horizontal refinement.

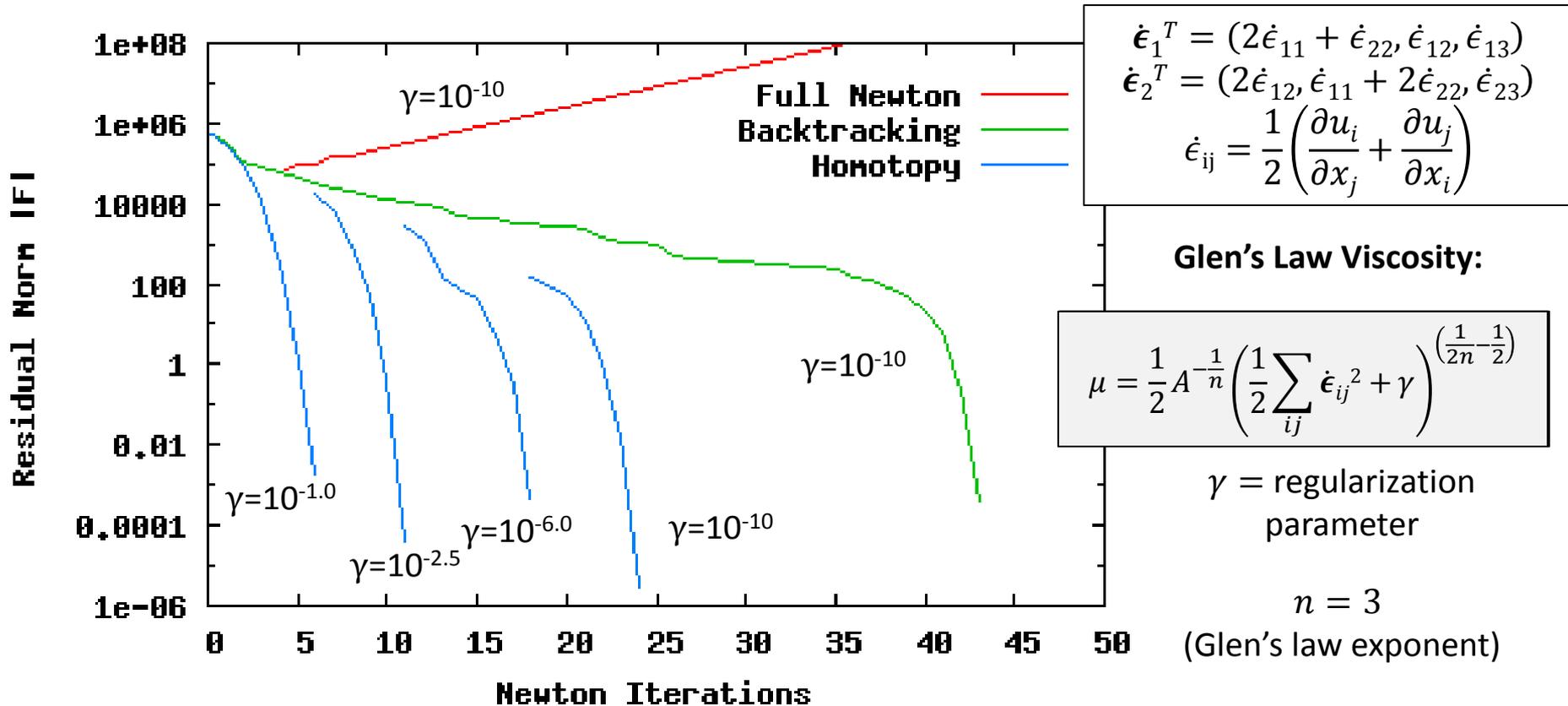
# Robustness of Newton's Method via Homotopy Continuation (LOCA)



# Robustness of Newton's Method via Homotopy Continuation (LOCA)



# Robustness of Newton's Method via Homotopy Continuation (LOCA)



- Newton's method most robust with full step + homotopy continuation of  $\gamma \rightarrow 10^{-10}$ : converges out-of-the-box!

# Scalability via Algebraic Multi-Grid Preconditioning

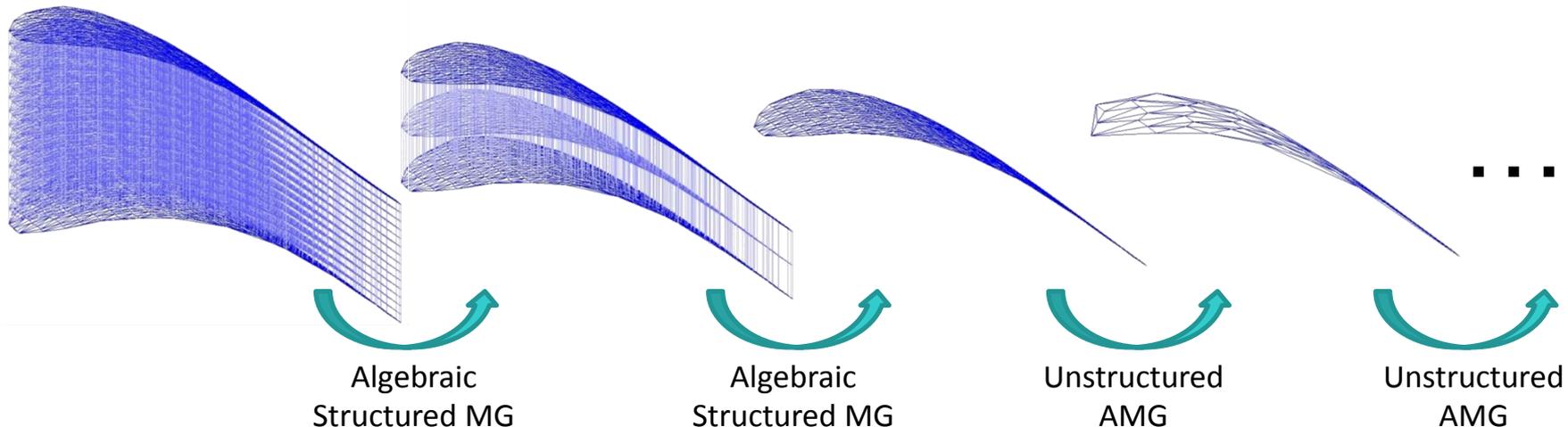
With R. Tuminaro (SNL)

Bad aspect ratios ruin classical AMG convergence rates!

- relatively small horizontal coupling terms, hard to smooth horizontal errors
- ⇒ Solvers (even ILU) must take aspect ratios into account

We developed a **new AMG solver** based on **semi-coarsening** (*figure below*)

- Algebraic Structured MG ( $\equiv$  matrix depend. MG) used with vertical line relaxation on finest levels + traditional AMG on 1 layer problem



\*With 2D partitioning and layer-wise node ordering, required for best performance of ILU.



# Scalability via Algebraic Multi-Grid Preconditioning

With R. Tuminaro (SNL)

Bad aspect ratios ruin classical AMG convergence rates!

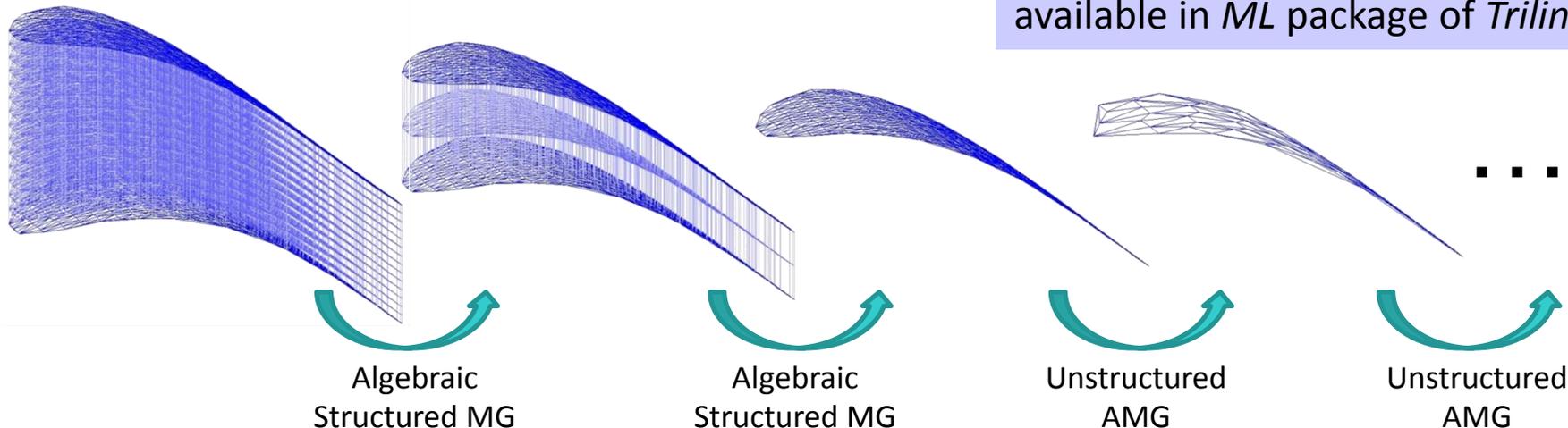
- relatively small horizontal coupling terms, hard to smooth horizontal errors
- ⇒ Solvers (even ILU) must take aspect ratios into account



We developed a **new AMG solver** based on **semi-coarsening** (*figure below*)

- Algebraic Structured MG ( $\equiv$  matrix depend. MG) used with vertical line relaxation on finest levels + traditional AMG on 1 layer problem

New AMG preconditioner is available in *ML* package of *Trilinos*!



\*With 2D partitioning and layer-wise node ordering, required for best performance of ILU.



# Scalability via Algebraic Multi-Grid Preconditioning

With R. Tuminaro (SNL)

Bad aspect ratios ruin classical AMG convergence rates!

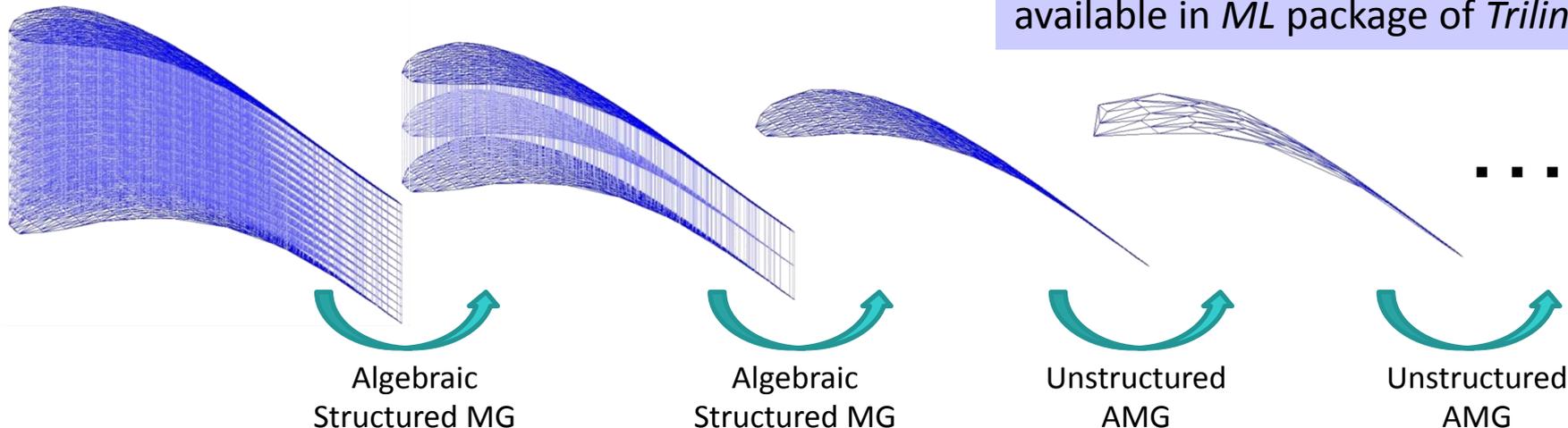
- relatively small horizontal coupling terms, hard to smooth horizontal errors
- ⇒ Solvers (even ILU) must take aspect ratios into account



We developed a **new AMG solver** based on **semi-coarsening** (*figure below*)

- Algebraic Structured MG ( $\equiv$  matrix depend. MG) used with vertical line relaxation on finest levels + traditional AMG on 1 layer problem

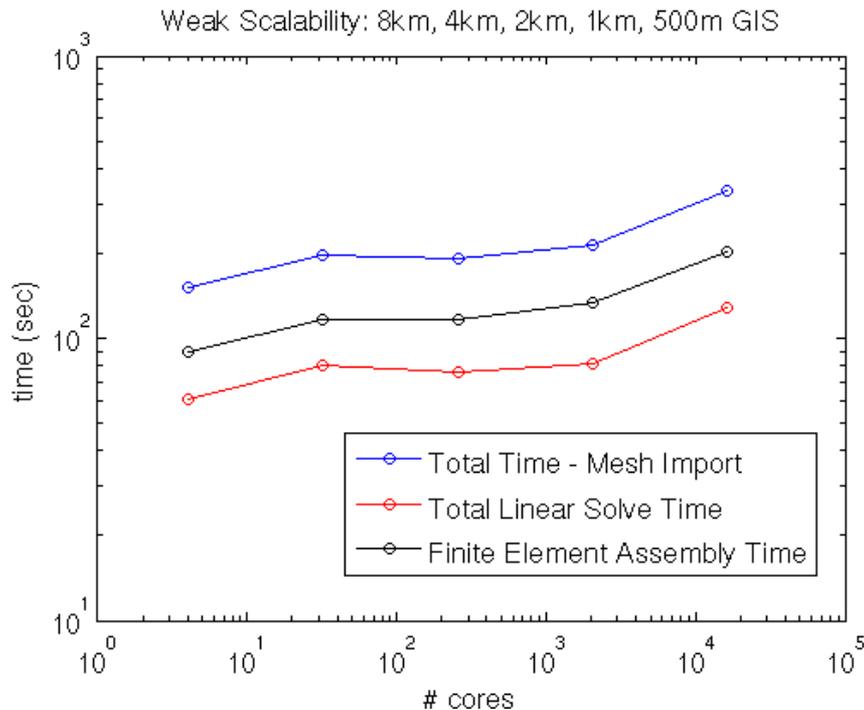
New AMG preconditioner is available in *ML* package of *Trilinos*!



\*With 2D partitioning and layer-wise node ordering, required for best performance of ILU.

***Scaling studies (next 3 slides):***  
New AMG preconditioner vs. ILU\*

# Greenland Controlled Weak Scalability Study



4 cores  
334K dofs  
8 km Greenland,  
5 vertical layers

$\times 8^4$   
scale up

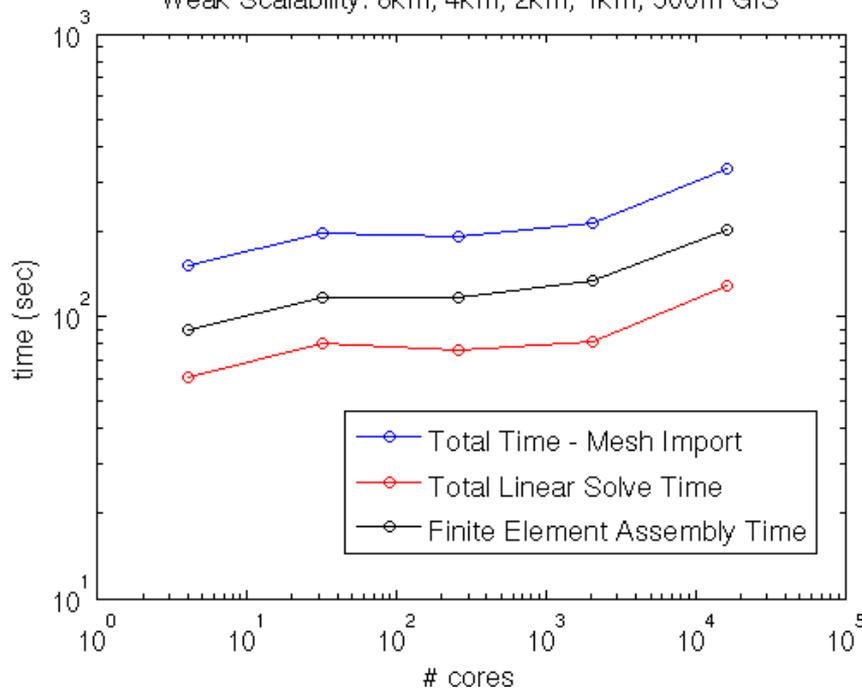
16,384 cores  
**1.12B dofs(!)**  
0.5 km Greenland,  
80 vertical layers

- Weak scaling study with fixed dataset, 4 mesh bisections.
- ~70-80K dofs/core.
- **Conjugate Gradient (CG) iterative method** for linear solves (faster convergence than GMRES).
- **New AMG preconditioner** developed by R. Tuminaro based on **semi-coarsening** (coarsening in z-direction only).
- **Significant improvement** in scalability with new AMG preconditioner over ILU preconditioner!

# Greenland Controlled Weak Scalability Study

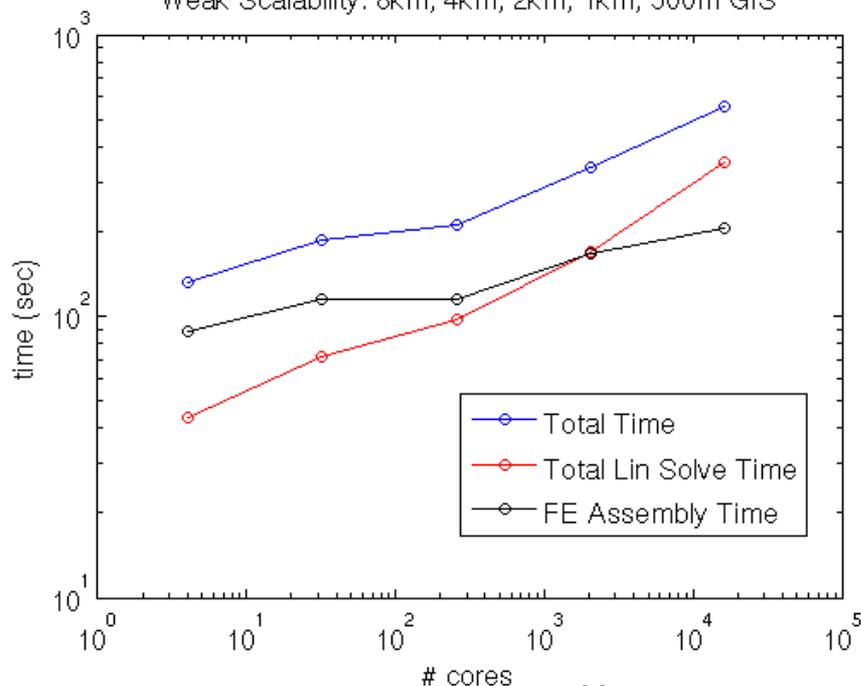
## New AMG preconditioner

Weak Scalability: 8km, 4km, 2km, 1km, 500m GIS



## ILU preconditioner

Weak Scalability: 8km, 4km, 2km, 1km, 500m GIS



4 cores  
334K dofs  
8 km Greenland,  
5 vertical layers

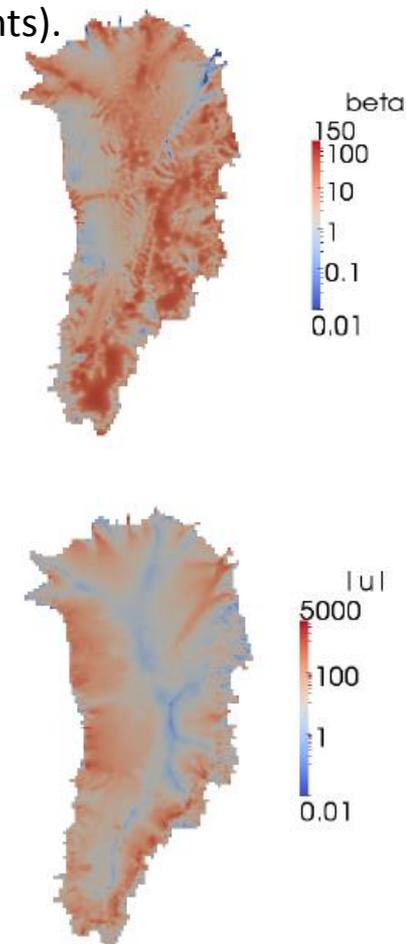
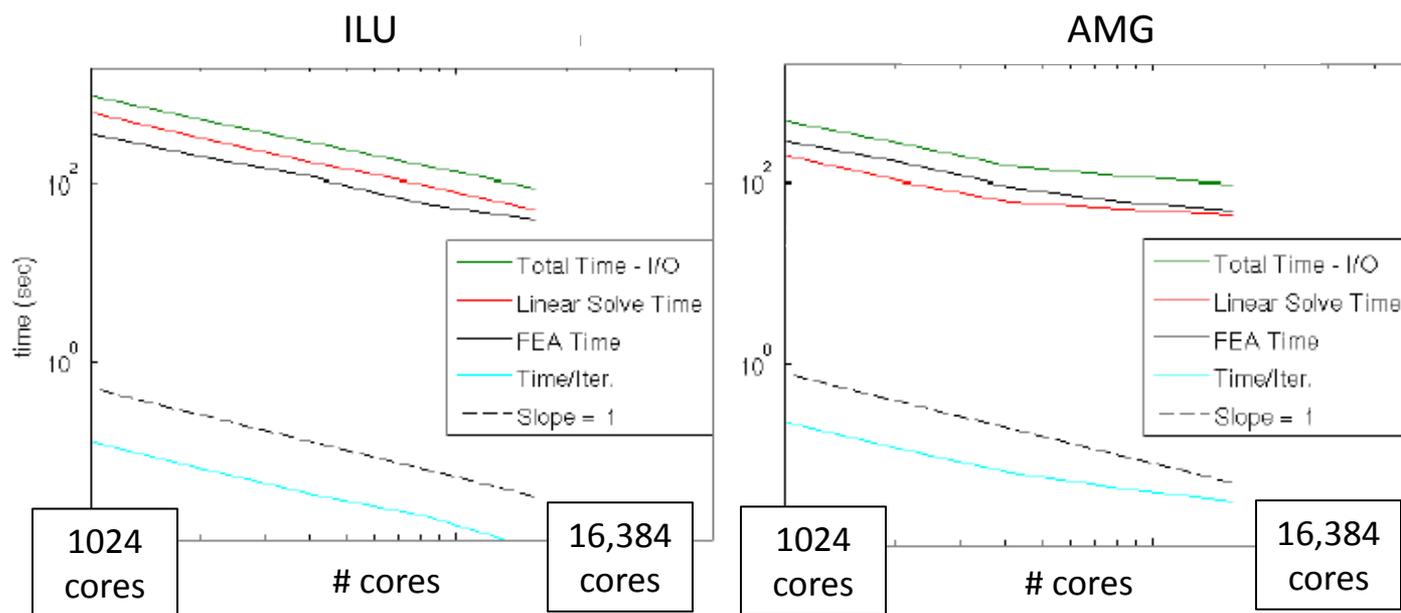
× 8<sup>4</sup>  
scale up

16,384 cores  
**1.12B dofs(!)**  
0.5 km Greenland,  
80 vertical layers

- **Significant improvement** in scalability with new AMG preconditioner over ILU preconditioner!

# Fine-Resolution Greenland Strong Scaling Study

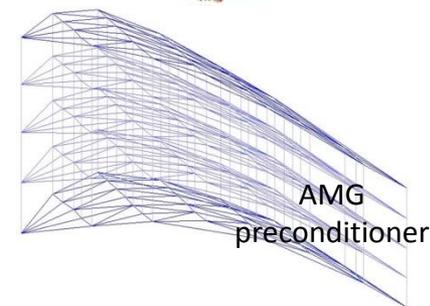
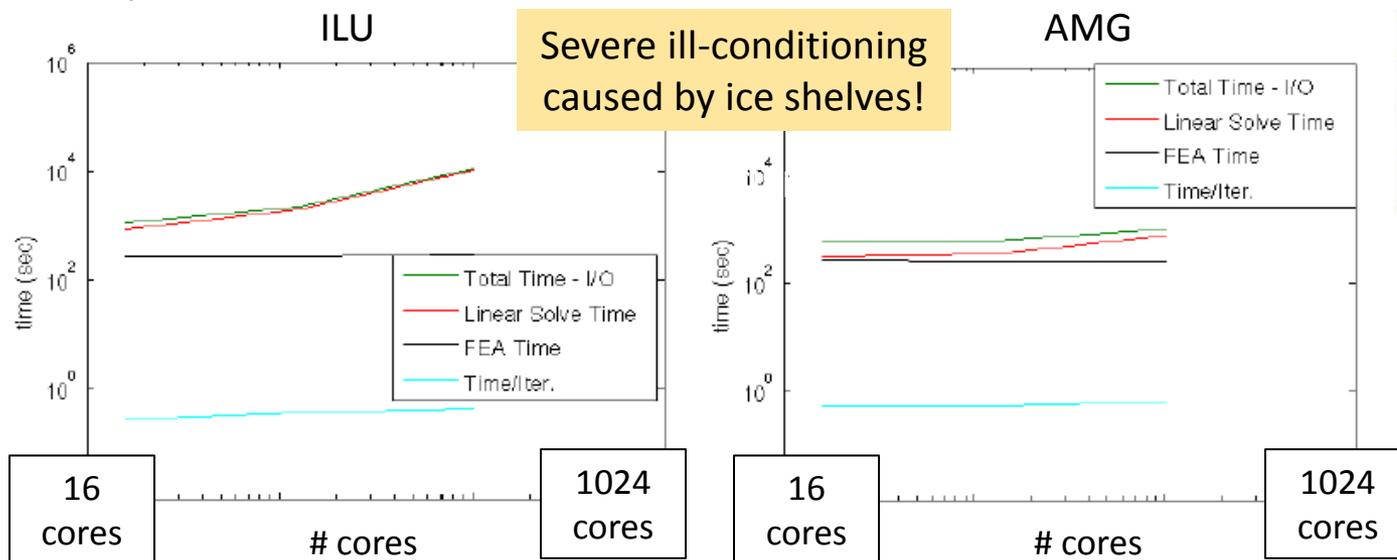
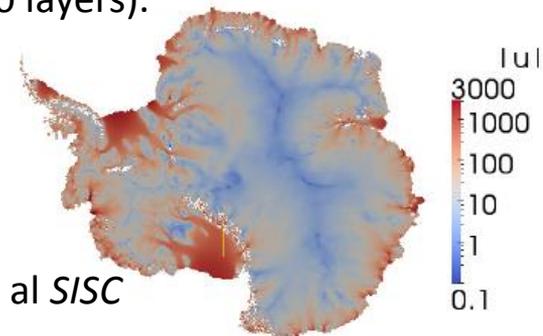
- Strong scaling on 1km Greenland with 40 vertical layers (143M dofs, hex elements).
- Initialized with realistic basal friction (from deterministic inversion) and temperature fields → interpolated from coarser to fine mesh.
- **Iterative linear solver:** CG.
- **Preconditioner:** ILU vs. new AMG (based on aggressive semi-coarsening).



ILU preconditioner scales better than AMG but ILU-preconditioned solve is slightly slower (see Kalashnikova et al *ICCS* 2015).

# Moderate Resolution Antarctica Weak Scaling Study

- Weak scaling study on Antarctic problem (8km w/ 5 layers → 2km with 20 layers).
- Initialized with realistic basal friction (from deterministic inversion) and temperature field from BEDMAP2.
- **Iterative linear solver:** GMRES.
- **Preconditioner:** ILU vs. new AMG based on aggressive semi-coarsening (Kalashnikova et al *GMD* 2014, Kalashnikova et al *ICCS* 2015, Tuminaro et al *SISC* 2015).



(vertical > horizontal coupling)  
+  
Neumann BCs  
=  
nearly singular submatrix associated with vertical lines

GMRES less sensitive than CG to rounding errors from ill-conditioning [also minimizes different norm].

AMG preconditioner less sensitive than ILU to ill-conditioning.

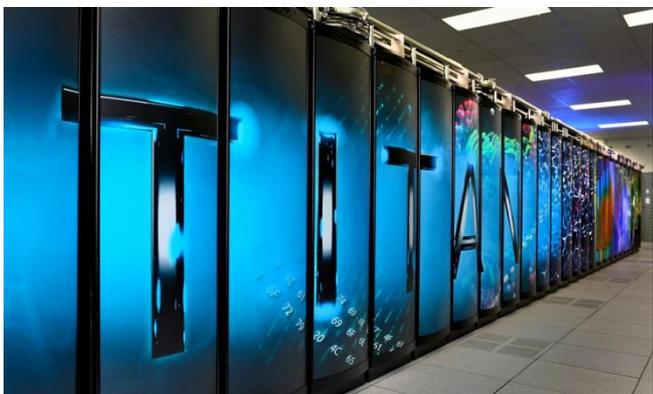
# Performance-Portability via *Kokkos*



With I. Demeshko (SNL)

We need to be able to run *Albany/FELIX* on **new architecture machines** (hybrid systems) and **manycore devices** (multi-core CPU, NVIDIA GPU, Intel Xeon Phi, etc.) .

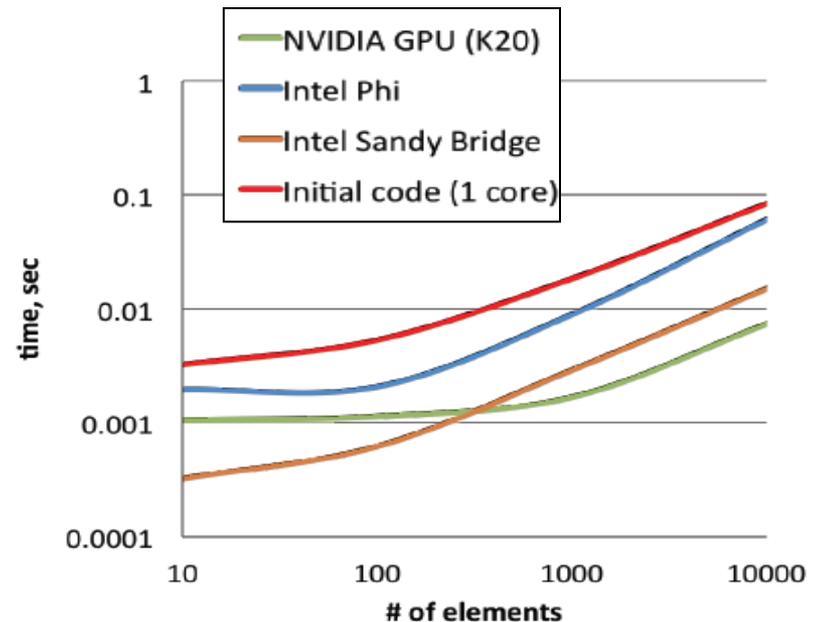
- ***Kokkos***: *Trilinos* library and programming model that provides performance portability across diverse devices with different memory models.
- With *Kokkos*, you write an algorithm once, and just change a template parameter to get the optimal data layout for your hardware.



See I. Demeshko's talk today @ 3:40PM in MS43  
"A *Kokkos* Implementation of *Albany*: A Performance Portable Multiphysics Simulation Code"

# Performance-Portability via Kokkos (continued)

- ***Right:*** results for a **mini-app** that uses finite element kernels from *Albany/FELIX* but none of the surrounding infrastructure.
  - “# of elements” = threading index (allows for on-node parallelism).
  - # of threads required before the Phi and GPU accelerators start to get enough work to warrant overhead: ~100 for the Phi and ~1000 for the GPU.
- ***Below:*** preliminary results for 3 of the finite element assembly kernels, as part of **full *Albany/FELIX*** code run.



Kernel	Serial	16 OpenMP Threads	GPU
Viscosity Jacobian	20.39 s	2.06 s	0.54 s
Basis Functions w/ FE Transforms	8.75 s	0.94 s	1.23 s
Gather Coordinates	0.097 s	0.107 s	5.77 s

**Note:** Gather Coordinates routine requires copying data from host to GPU.

# Summary and Ongoing Work

---

## Summary:

- This talk described the development of a finite element land ice solver known as *Albany/FELIX* written using the libraries of the *Trilinos* libraries.
- The code is verified, scalable, robust, and portable to new-architecture machines! This is thanks to:
  - Some new algorithms (e.g., AMG preconditioner) and numerical techniques (e.g., homotopy continuation).
  - The *Trilinos* software stack.

Use of *Trilinos* libraries has enabled the rapid development of this code!

## Ongoing/future work:

- Dynamic simulations of ice evolution.
- Deterministic and stochastic initialization runs (see M. Perego's talk).
- Porting of code to new architecture supercomputers (see I. Demeshko's talk).
- Articles on *Albany/FELIX* [*GMD, ICCS 2015*], *Albany* [*J. Engng.*] (see A. Salinger's talk), AMG preconditioner (*SISC*).
- Delivering code to climate community and coupling to earth system models.

# Funding/Acknowledgements

Support for this work was provided through Scientific Discovery through Advanced Computing (**SciDAC**) projects funded by the U.S. Department of Energy, Office of Science (**OSCR**), Advanced Scientific Computing Research and Biological and Environmental Research (**BER**) → **PISCEES SciDAC Application Partnership.**



***PISCEES team members:*** W. Lipscomb, S. Price, M. Hoffman, A. Salinger, M. Perego, I. Kalashnikova, R. Tuminaro, P. Jones, K. Evans, P. Worley, M. Gunzburger, C. Jackson;  
***Trilinos/DAKOTA collaborators:*** E. Phipps, M. Eldred, J. Jakeman, L. Swiler.

***Thank you! Questions?***

# References

---

- [1] M.A. Heroux *et al.* "An overview of the Trilinos project." *ACM Trans. Math. Softw.* **31**(3) (2005).
- [2] A.G. Salinger *et al.* "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", *Comput. Sci. Disc.* (submitted, 2015).
- [3] **I. Kalashnikova**, M. Perego, A. Salinger, R. Tuminaro, S. Price. "Albany/FELIX: A Parallel, Scalable and Robust Finite Element Higher-Order Stokes Ice Sheet Solver Built for Advanced Analysis", *Geosci. Model Develop. Discuss.* 7 (2014) 8079-8149 (under review for *GMD*).
- [4] **I. Kalashnikova**, R. Tuminaro, M. Perego, A. Salinger, S. Price. "On the scalability of the Albany/FELIX first-order Stokes approximation ice sheet solver for large-scale simulations of the Greenland and Antarctic ice sheets", *MSESM/ICCS15*, Reykjavik, Iceland (June 2014).
- [5] R.S. Tuminaro, **I. Tezaur**, M. Perego, A.G. Salinger. "A Hybrid Operator Dependent Multi-Grid/Algebraic Multi-Grid Approach: Application to Ice Sheet Modeling", *SIAM J. Sci. Comput.* (in prep).